DEFENSIVE PUBLICATION

UNITED STATES PATENT OFFICE

Published at the request of the applicant or owner in accordance with the Notice of Dec. 16, 1969, 869 O.G. 687. The abstracts of Defensive Publication applications are identified by distinctly numbered series and are arranged chronologically. The heading of each abstract indicates the number of pages of specification, including claims and sheets of drawings contained in the application as originally filed. The files of these applications are available to the public for inspection and reproduction may be purchased for 30 cents a sheet.

Defensive Publication applications have not been examined as to the merits of alleged invention. The Patent Office makes no assertion as to the novelty of the disclosed subject matter.

PUBLISHED MAY 7, 1974

922 O.G. 10

T922,008

DATA PROCESSOR PROGRAM FOR DETERMINING CHARACTER STRING SIMILARITY

Donald C. Gause, Owego, and Eduardo Kellerman and Gerald L. Rouse, Endicott, N.Y., assignors to International Business Machines Corporation, Armonk, N.Y.

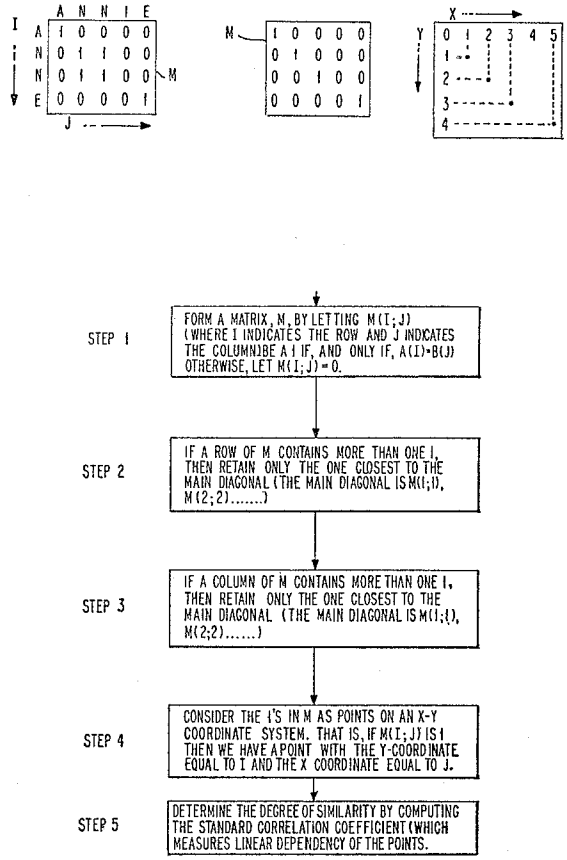
Continuation of application Ser. No. 154,093, June 17, 1971. This application Oct. 18, 1973, Ser. No. 407,733

Int. Cl. G05b *19/28;* G06f *7/34*

U.S. Cl. 444—1

4 Sheets Drawing. 27 Pages Specification

An improved program method is shown for determining the similarity of two character strings where they are not necessarily equal character by character. Each character of one character string is compared with each character of the other character string, and the results are stored in a matrix. Each row of the matrix corresponds to a respective one of the characters of the first string, and each column of the matrix corresponds to a respective character of the other string. A logical one value is stored in each matrix position where corresponding characters of the first and second string are equal, and a logical zero is stored in each position where they are not equal. In the event that there is more than one logical one in each row (or in each column, or both), all of the logical ones in that row (or column, or both) except that which is closest to the major diagonal of the matrix are changed to logical zero. In the event that two logical ones in a row (or column, or both) are equidistant from the major diagonal, both may be retained. The remaining logical ones in the matrix are then considered as points on an X, Y coordinate system, and the standard correlation coefficient (which measures linear dependence) of the points is calculated to determine the degree of similarity between the two character strings. A decision that sufficient similarity exists to make a determination that a match has occurred may be based upon various known tests, for example, exceeding predetermined thresholds; or, in the event that a match is made against several character strings, the highest coefficient calculated can be determined as the match. The improved method can be used in programming areas such as inquiry (or query) systems and the job control language processor of an operating system. The program usually handles missing characters, transposed characters and other common errors. The program can be either a fixed or an adaptive type.



May 7, 1974 D. C. GAUSE ET AL T922,008
DATA PROCESSOR PROGRAM FOR DETERMINING CHARACTER STRING SIMILARITY
Original Filed June 17, 1971 4 Sheets-Sheet 1

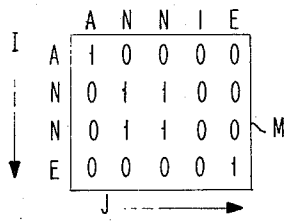


FIG. 2a

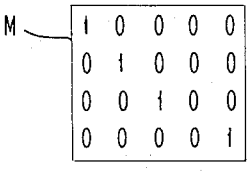


FIG. 2b

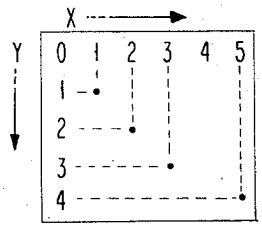
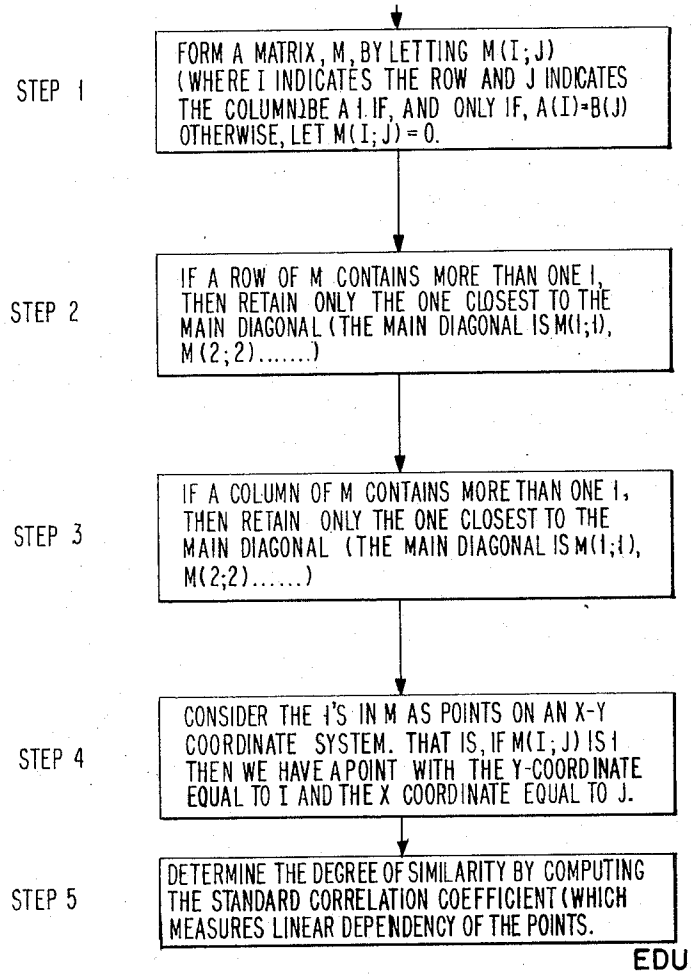


FIG. 2c

FIG. 1



INVENTORS
EDUARDO KELLERMAN
GERALD L. ROUSE
DONALD C. GAUSE

BY John C. Black
ATTORNEY

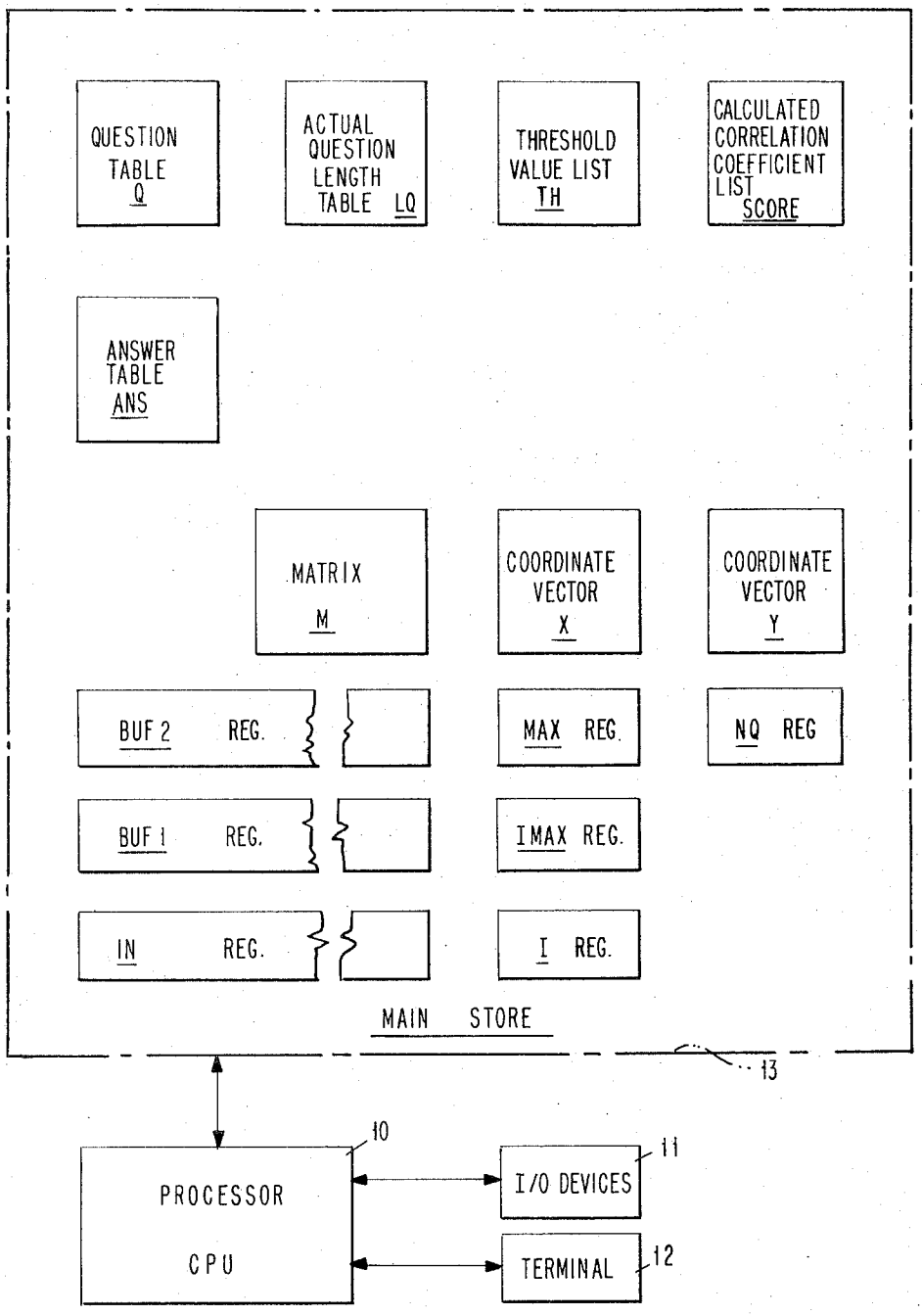

QUESTION TABLE Q
ACTUAL QUESTION LENGTH TABLE LQ
THRESHOLD VALUE LIST TH
CALCULATED CORRELATION COEFFICIENT LIST SCORE
ANSWER TABLE ANS
MATRIX M
COORDINATE VECTOR X
COORDINATE VECTOR Y
BUF 2 REG.
MAX REG.
NQ REG
BUF 1 REG.
IMAX REG.
IN REG.
I REG.
MAIN STORE
13
10
PROCESSOR
CPU
I/O DEVICES
11
TERMINAL
12

FIG. 3

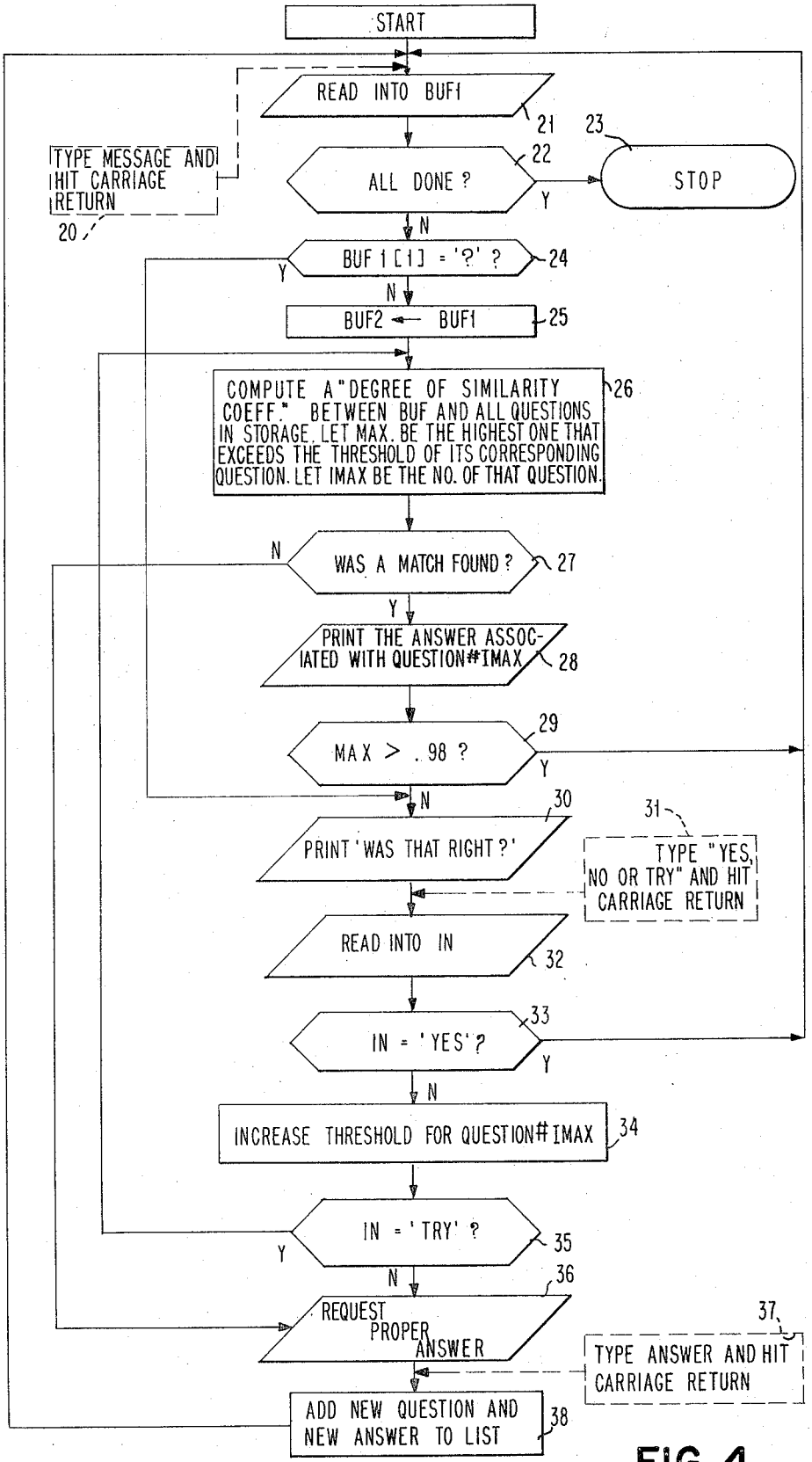

START
READ INTO BUF1
21
23
TYPE MESSAGE AND HIT CARRIAGE RETURN
20
22
ALL DONE ?
STOP
Y
N
BUF1[1] = '?' ?
24
Y
N
BUF2 ← BUF1
25
COMPUTE A "DEGREE OF SIMILARITY COEFF." BETWEEN BUF AND ALL QUESTIONS IN STORAGE. LET MAX. BE THE HIGHEST ONE THAT EXCEEDS THE THRESHOLD OF ITS CORRESPONDING QUESTION. LET IMAX BE THE NO. OF THAT QUESTION.
26
N
WAS A MATCH FOUND ?
27
Y
PRINT THE ANSWER ASSOCIATED WITH QUESTION#IMAX
28
29
MAX > .98 ?
Y
N
30
31
PRINT 'WAS THAT RIGHT ?'
TYPE "YES, NO OR TRY" AND HIT CARRIAGE RETURN
READ INTO IN
32
33
IN = 'YES' ?
Y
N
INCREASE THRESHOLD FOR QUESTION#IMAX
34
IN = 'TRY' ?
35
Y
N
36
REQUEST PROPER ANSWER
37
TYPE ANSWER AND HIT CARRIAGE RETURN
ADD NEW QUESTION AND NEW ANSWER TO LIST
38

FIG. 4

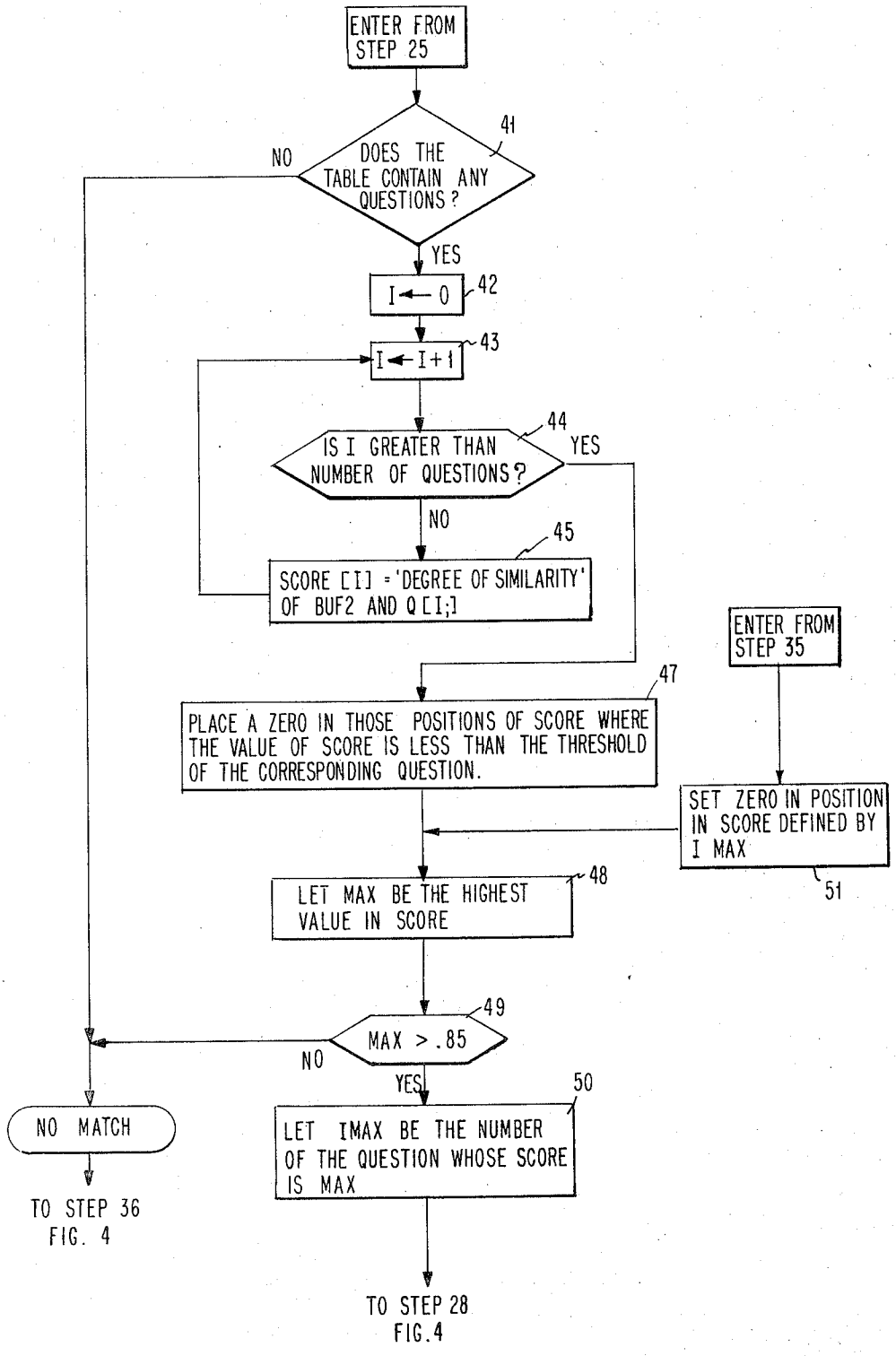

ENTER FROM STEP 25
41
DOES THE TABLE CONTAIN ANY QUESTIONS?
NO
YES
I ← 0
42
I ← I+1
43
IS I GREATER THAN NUMBER OF QUESTIONS?
44
YES
NO
45
SCORE [I] = 'DEGREE OF SIMILARITY' OF BUF2 AND Q [I;]
ENTER FROM STEP 35
47
PLACE A ZERO IN THOSE POSITIONS OF SCORE WHERE THE VALUE OF SCORE IS LESS THAN THE THRESHOLD OF THE CORRESPONDING QUESTION.
SET ZERO IN POSITION IN SCORE DEFINED BY I MAX
51
LET MAX BE THE HIGHEST VALUE IN SCORE
48
49
MAX > .85
NO
YES
50
NO MATCH
LET IMAX BE THE NUMBER OF THE QUESTION WHOSE SCORE IS MAX
TO STEP 36 FIG. 4
TO STEP 28 FIG. 4

FIG. 5